

# Ανάπτυξη και Σχεδίαση Λογισμικού

Η γλώσσα  
προγραμματισμού C

Γεώργιος Δημητρίου

# Συναρτήσεις (Functions)

- Οι βασικές λειτουργικές ενότητες ενός προγράμματος C
- Καλούνται με ορίσματα που αντιστοιχούνται σε ειδικές τοπικές μεταβλητές, τις τυπικές παραμέτρους, εκτελούν τον κώδικά τους και επιστρέφουν στο σημείο κλήσης τους
- Έχουν άμεσα ή έμμεσα αποτελέσματα
- Μπορούν να καλούνται αναδρομικά

# Συναρτήσεις της C – Τα Βασικά

- Δήλωση συνάρτησης
- Παράμετροι και αποτέλεσμα
- Κλήση συνάρτησης

# Δήλωση (Ορισμός) Συνάρτησης

- Ο τύπος και το όνομα της συνάρτησης
  - Ο τύπος είναι ο τύπος αποτελέσματος
- Οι παράμετροι της συνάρτησης
- Το σώμα της συνάρτησης
  - Στην ουσία μία σύνθετη εντολή

# Τύπος Συνάρτησης

- Αν η συνάρτηση δεν επιστρέφει κάποια τιμή, ο τύπος της είναι void
- Διαφορετικά είναι ο τύπος του αποτελέσματος που επιστρέφεται άμεσα από τη συνάρτηση
  - Άμεση επιστροφή με εντολή return
- Δεν επιτρέπεται επιστροφή πίνακα (αλλά επιτρέπεται επιστροφή δείκτη)

# Παράμετροι Συναρτήσεων

- Μια λίστα από μεταβλητές με τον τύπο καθεμιάς ξεχωριστά μέσα σε παρένθεση
  - Σε κάθε παράμετρο αντιστοιχείται κάποιο όρισμα κατά την κλήση της συνάρτησης
- Μια συνάρτηση μπορεί να μην έχει παραμέτρους
  - Μπορούμε να αφήσουμε τη λίστα κενή, ή να βάλουμε την ένδειξη void

# Παραδείγματα

```
int fu1 (int x, int y) {  
    return x*y;  
}  
void fu2(int n, int *a, int *b, int *c) {  
    int i;  
    for (i = 0; i < n; i++) c[i] = fu1 (a[i],b[i]);  
}  
void done(void) {  
    printf("Done!\n");  
}
```

επικεφαλίδα συνάρτησης

σώμα συνάρτησης

Ανάπτυξη και Σχεδίαση Λογισμικού  
Η γλώσσα προγραμματισμού C

# Τοπικές Μεταβλητές

- Οι μεταβλητές που δηλώνονται μέσα στο σώμα της συνάρτησης
- Τοπικές σημαίνει ότι δεν μπορούν να τις δουν άλλες συναρτήσεις
- Πιθανή αρχικοποίησή τους μετατρέπεται σε ανάθεση



# Καθολικές Μεταβλητές

- Μεταβλητές που δηλώνονται εκτός συναρτήσεων
- Τις βλέπουν όλες οι συναρτήσεις
  - Εκτός αν επισκιάζονται από τοπικές μεταβλητές με το ίδιο όνομα
- Πιθανή αρχικοποίησή τους γίνεται με την φόρτωση του προγράμματος

# Εμβέλεις

- Φωλιασμένες δηλώσεις μεταβλητών σε σύνθετες εντολές
- Ένα όνομα επισκιάζει το ίδιο όνομα από εξωτερική εμβέλεια
- Στο τέλος της σύνθετης εντολής επανερχόμαστε στα ονόματα της προηγούμενης εμβέλειας
- Όμοια για τις τοπικές σε σχέση με τις καθολικές μεταβλητές

# Κλήση Συνάρτησης

- Μια συνάρτηση καλείται με το όνομά της και σε παρένθεση εκφράσεις – ορίσματα που αντιστοιχούνται στις παραμέτρους με την ίδια σειρά
- Ο έλεγχος μεταβαίνει στο σώμα της συνάρτησης και το εκτελεί χρησιμοποιώντας τις τιμές των εκφράσεων ως αρχικές τιμές των παραμέτρων
- Με το τέλος της συνάρτησης, ο έλεγχος επιστρέφει στο σημείο που ακολουθεί το σημείο κλήσης

# Πέρασμα Παραμέτρων

- Η χρήση των τιμών των ορισμάτων για τις παραμέτρους υπονοεί «πέρασμα κατ' αξία»
- «Πέρασμα κατ' αναφορά» υλοποιείται μέσω δεικτών
  - Ο δείκτης ως μεταβλητή έχει τιμή μια διεύθυνση
  - Αν γράψουμε εκεί που δείχνει ο δείκτης, στην ουσία γράφουμε στο στοιχείο της καλούσας συνάρτησης
  - Οι πίνακες περνάνε πάντα ως δείκτες, άρα κατ' αναφορά

# Επιστροφή Αποτελέσματος

- Με την εντολή `return` το άμεσο αποτέλεσμα της συνάρτησης επιστρέφει στο καλούν περιβάλλον
  - Αυτό σημαίνει ότι η κλήση γίνεται μέσα σε έκφραση
  - Εντολή `return` μπορεί να υπάρχει οπουδήποτε στο σώμα της συνάρτησης
    - Και για συναρτήσεις χωρίς άμεσο αποτέλεσμα!
- Έμμεσα αποτελέσματα λαμβάνονται με το πέρασμα κατ' αναφορά ή με καθολικές μεταβλητές

# Παράδειγμα

```
int main() {  
    int x[100], y[100], z[100];  
    fu2 (100, x, y, z);  
    done ();  
}
```

# Παράδειγμα

```
int main() {  
    int x[100], y[100], z[100];  
    fu2 (100, x, y, z);  
    done ();  
}
```

# Παράδειγμα

```
int main() {  
    int x[100], y[100], z[100];  
    fu2 (100, x, y, z);  
    done ();  
}
```

```
void fu2 (int n, int *a, int *b, int *c) {  
    ...  
    for ... fu1 (a[i],b[i]);  
}
```



# Παράδειγμα

```
int main() {  
    int x[100], y[100], z[100];  
    fu2 (100, x, y, z);  
    done ();  
}
```

```
void fu2 (int n, int *a, int *b, int *c) {  
    ...  
    for ... fu1 (a[i],b[i]);  
}
```

# Παράδειγμα

```
int main() {  
    int x[100], y[100], z[100];  
    fu2 (100, x, y, z);  
    done ();  
}
```

```
void fu2 (int n, int *a, int *b, int *c) {  
    ...  
    for ... fu1 (a[i],b[i]);  
}
```

a[0],b[0]

```
int fu1 (int x, int y) {  
    return x*y;  
}
```

# Παράδειγμα

```
int main() {  
    int x[100], y[100], z[100];  
    fu2 (100, x, y, z);  
    done ();  
}
```

```
void fu2 (int n, int *a, int *b, int *c) {  
    ...  
    for ... fu1 (a[i],b[i]);  
}
```

# Παράδειγμα

```
int main() {  
    int x[100], y[100], z[100];  
    fu2 (100, x, y, z);  
    done ();  
}
```

```
void fu2 (int n, int *a, int *b, int *c) {  
    ...  
    for ... fu1 (a[i],b[i]);  
}
```

a[1],b[1]

```
int fu1 (int x, int y) {  
    return x*y;  
}
```

# Παράδειγμα

```
int main() {  
    int x[100], y[100], z[100];  
    fu2 (100, x, y, z);  
    done ();  
}
```

```
void fu2 (int n, int *a, int *b, int *c) {  
    ...  
    for ... fu1 (a[i],b[i]);  
}
```

# Παράδειγμα

```
int main() {  
    int x[100], y[100], z[100];  
    fu2 (100, x, y, z);  
    done ();  
}
```

```
void fu2 (int n, int *a, int *b, int *c) {  
    ...  
    for ... fu1 (a[i],b[i]);  
}
```

KOK

# Παράδειγμα

```
int main() {  
    int x[100], y[100], z[100];  
    fu2 (100, x, y, z);  
    done ();  
}
```

```
void fu2 (int n, int *a, int *b, int *c) {  
    ...  
    for ... fu1 (a[i],b[i]);  
}
```

# Παράδειγμα

```
int main() {  
    int x[100], y[100], z[100];  
    fu2 (100, x, y, z);  
    done ();  
}
```

```
void fu2 (int n, int *a, int *b, int *c) {  
    ...  
    for ... fu1 (a[i],b[i]);  
}
```



# Παράδειγμα


```
int main() {  
    int x[100], y[100], z[100];  
    fu2 (100, x, y, z);  
    done ();  
}
```

# Παράδειγμα

```
int main() {  
    int x[100], y[100], z[100];  
    fu2 (100, x, y, z);  
    done ();  
}
```

# Παράδειγμα

```
int main() {  
    int x[100], y[100], z[100];  
    fu2 (100, x, y, z);  
    done ();  
}
```



```
void done(void) {  
    printf("Done!\n");  
}
```

# Παράδειγμα

```
int main() {  
    int x[100], y[100], z[100];  
    fu2 (100, x, y, z);  
    done ();  
}
```

# Παράδειγμα

```
int main() {  
    int x[100], y[100], z[100];  
    fu2 (100, x, y, z);  
    done ();  
}
```

# Τι Μάθαμε Σήμερα

- Συναρτήσεις της C
- Παράμετροι και πέρασμά τους
- Τοπικές/Καθολικές μεταβλητές – Εμβέλεια